A Personalized Forum Topic Ranking System Based on Maximum Entropy Model

Author: Sen Li[1]

Instructor: Shi-Min Wei[2], Hong-Jie Jia[1], Ying Gao[1]

[1]Beijing No.8 Middle School

[2]School of Automation, Beijing University of Posts and Telecommunications

---

Abstract

Forums have been one of the most important internet services since the 21st century. However, Forum users have to receive information in a passive way currently. The topics in forums are ordered by last update time (last reply time), thus the information that a user is interested in may be overwhelmed by a large number of other information. Users always have to scan many pages to find a minority of information they need. In this paper, based on the analysis of users' needs, I have designed a personalized forum topic ranking system. This personalized ranking system first calculates all the factors that will influence the user's decision-making as to whether or not to view a topic by using his/her browsing history. Then the system predicts the click probability for each topic according to all the influence factors using a learned maximum entropy model. Finally, forum topics can be ranked by the predicted click probability, so as to make users find their favorite information easier. As shown by the experiments, the precision of the personalized ranking system is about 60% to 75%, which improves the traditional method (ordered by last update time) by 50% to 85%. In addition, with the normalization of the indicator functions of the maximum entropy model and the selection of the iterative endpoint, the training time can be lowered to an average of 0.01 seconds for each user. It indicates that such a model is able to meet the requirements of practical applications.

Innovations

1) First personalized ranking system for forum topics (There are some previous works for news and commodities)
2) First maximum entropy model based personalized system (Use maximum entropy model to integrate the influence factors)
3) Using multiple influence factors to describe the user's preferences (Previous works usually only consider the influence of the article contents or use collaborative recommendation algorithm)
4) Handling users' multiple interest areas and new interest areas using vector dimension selection (Previous works usually use clustering method or forgetting algorithm)
5) Improving the training speed of the maximum entropy model with iterative endpoint selection
6) Improving the training speed of the maximum entropy model with indicator function normalization

According to the novelty retrieval performed by China Machinery Industry Information Institute, which was authorized as the national level Sci-tech novelty retrieval institution, there is no previous work on section 1 to 5 in the world. Section 6 was proposed after the novelty retrieval. According the retrieval performed by myself, there is no previously published papers on section 6.

1. Introduction

1.1 Background of Personalized Forum Topic Ranking

According to the statistics from China Internet Network Information Center [1], there are 338 million netizens in China, and 43.2% of them often use forums (as of June 2009). It shows that forums have become one of the most important internet services. Yet, at present, forum systems build topic lists with topics ordered by last update time (last reply time). Then, they show the list to all the forum users. As different users have different preferences, topics needed by each user are not the same. Under the current topic list ranking scheme, the information that a user is interested in may be overwhelmed by a large number of other information. Users always have to scan many pages to find a minority of information they need. Especially in some large forums, when someone posts a new topic that only interests a few users, it will be soon pushed beyond the first 10 pages within one or two days, due to the topic is not attractive to other users. Such a condition makes users miss many opportunities to access the information they are interested in.

The investigation by Xiao-Ming Li [2] has analyzed almost 500,000 click-throughs from Tianwang, which is a large-scale general search engine in China. He found that in the search results lists, there are 47.3% of the click-throughs occurred in the first pages and 75.6% of them occurred in the first five pages. Thus, it shows that only a few users will scan many pages in the search results list, and most of the users will only focus on the results in the first few pages. According to my experience and the feedback from other users, although there are no relevant investigations in forum systems, the situation in forums are similar. Consequently, users will miss many topics they may interest in, but with an earlier last update time.

The current topic list ranking scheme has greatly reduced people's efficiency when obtaining information from forums. People prefer to browse forums in a more convenient way, which forum systems can provide the information they need automatically. And each user can get a topic list based on his/her preferences so as to make every forum user easier to find the information he/she wants.

1.2 Definition of the Personalized Forum Topic Ranking

This personalized forum topic ranking system will first builds the preferences data for each user by using his/her browsing history. Then the system will display a personalized topic list for each user, which makes the topics ordered in accordance with the user's preferences. That is, based on the prediction by the ranking system, topics meet the user's preferences better will have higher ranks in the topic list.

1.3 Previous Work on Personalized Browsing System

According to previously published papers, there is no existing research on personalized forum topic browsing system. Nevertheless, there are plenty of work on the personalized web page browsing. However, most of the work [3~7] only considered the influence of the article contents on the users' browsing preferences, without taking account of other influence factors. The shortcomings of these works are that many factors will influence users' browsing preferences. For example, a user may like to view a topic, as the topic is posted by his/her favorite author or the author is very famous in the forum, although the topic is not in the user’s interest area. In some other studies [8][9], the system will recommend some articles focused by other users who have similar browsing behavior to the current

user. However, sometimes the user may not interested in the articles, as the recommendation is made without considering the user's own preferences.

1.4 Previous Work on Maximum Entropy Model

Maximum entropy model was established by E. T. Jaynes in 1957. In 1992, D.Pietra introduced it into natural language processing [10]. At present, maximum entropy model has been widely used in many areas such as phrase recognition, syntactic parsing, POS tagging and text classification. The maximum entropy model can integrate all the factors affecting on the target event, without taking into account the structure of the factors. Weights of the factors can be obtained throughout the training. After training, the occurrence probability of the target event under the influence of all the factors can be obtained. Such a perfect mathematical attribute makes the maximum entropy model have a wonderful performance in a variety of areas [10]. For example, in spam filtering [11], the maximum entropy model is significantly better than other methods such as SVM, Bayesian, and decision tree. In the text classification [12], the maximum entropy model is significantly better than the Bayesian method, and has similar performance to the SVM method. Nonetheless, maximum entropy model has not been used in personalized browsing according to previously published papers.

Although the maximum entropy model has a perfect performance, its practical application has been greatly limited due to the tremendously slow training speed. Many innovative training algorithms have been described in order to improve the training speed, but the training speed is still too low to meet the requirements of practical applications. In addition, the indicator function normalization and the iterative endpoint selection have not been introduced to improve the training speed in the previously published papers.

1.5 Overview of the Work in This Paper

In order to describe users' preferences comprehensively, I have proposed 10 factors that will influence users' browsing preferences based on the analysis of users' browsing habit. Moreover, I have established the calculation methods of the influence factors' occurrence probabilities. Bayesian network or maximum entropy model is commonly used to calculate the occurrence probability of the target event under the effects of multiple factors. Since the performance of the maximum entropy model is better than the Bayesian network in theory and in practice in other areas, we choose to use the maximum entropy model to integrate the 10 influence factors.

This personalized ranking system first calculates all the factors that will influence the user's decision-making as to whether or not to view a topic by using his/her browsing history. Then the system predicts the click probability for each topic according to all the influence factors using a learned maximum entropy model. Finally, forum topics can be ranked by the predicted click probability. The architecture of the personalized ranking system is as follows:

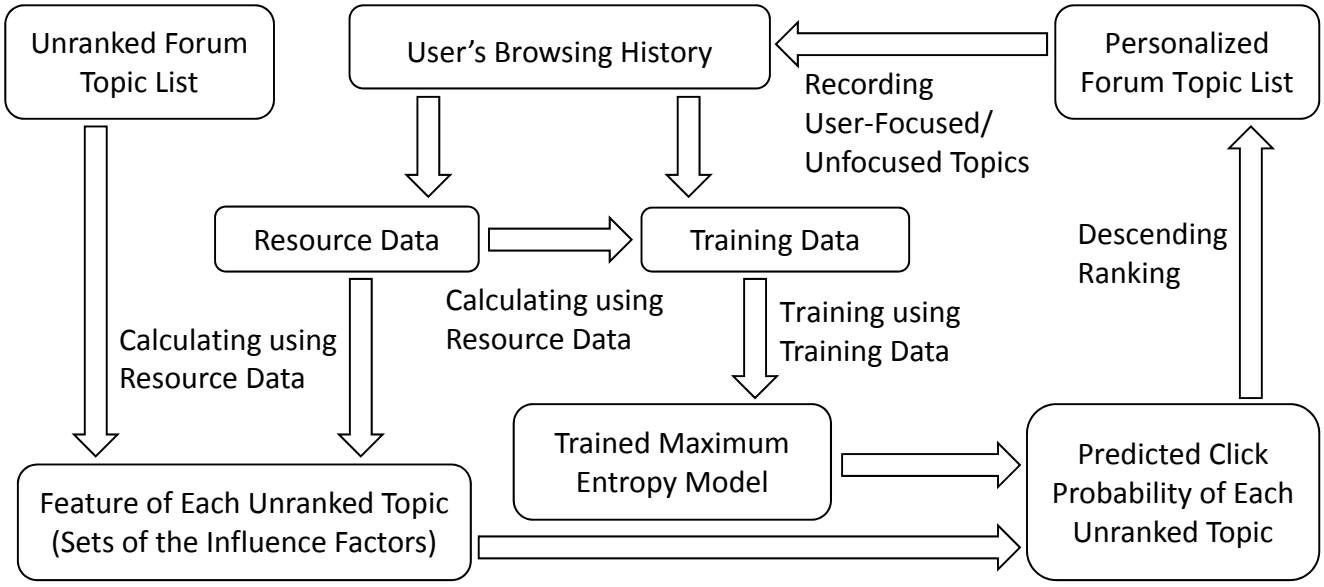


Figure 1: Architecture of the Personalized Forum Topic Ranking System

In order to handle users' multiple interest areas and new interest areas, I have proposed using vector dimension selection to supersede the clustering method, which has been widely used previously. In the experiment part, I have drawn a comparison of the precision between the personalized ranking system and the traditional ranking scheme. I have evaluated the effects of the vector dimension selection and the user's browsing history amount.

In addition, with the aim of improving the training speed of the maximum entropy model, I have optimized the SCGIS algorithm and proposed using the indicator function normalization and the iterative endpoint selection to accelerate the training process. Moreover, in the experiments, I have evaluated the effects of these two methods.

2. Building Training Data and Resource Data

In order to provide personalized forum topic lists for the user, user's browsing history is needed to be collected. The browsing history record is divided into two parts: 1. User-focused topic list; 2. User-unfocused topic list. Subsequently, the training data and the resource data can be obtained from the browsing history record.

2.1 Building User-Focused/Unfocused Topic List

First of all, we need to define the users' behavior. If a user replied a topic, he/she is considered to have focused on the topic. If a user has viewed a topic over a specified time, it can also be considered that the user has focused on the topic. Moreover, users should also be able to choose to focus on a topic. Using user-focused topic list instead of user-clicked topic list is aimed to avoid the adverse effect of wrong clicks. For example, a user may find he/she does not like the topic once he/she clicked on the tittle of the topic.

When a user has focused on a topic posted by him/herself, many occurrence probabilities of the factors in the topic feature will be constant equal to one. Therefore, we do not put a topic posted by the user into his/her user-focused/unfocused topic list.

In the experiment part, since users are not able to choose to focus on a topic in the current forum system and the current forum system also not able to record users' viewing time, we will only regard a topic as user-focused after the user has replied the topic, and the user-focused time is defined as

the replied time. Thus, user-focused topic list can be obtained from the replies database or collected whenever a user first replies a topic.

When a topic is not focused by the user, it does not always indicates the user is not interested in the topic. Instead, the user may not have noticed the topic or he/she is planning to view the topic in the future. Therefore, we use the following method to obtain the user-unfocused topic list:

1) If a user has focused on a topic, therefore we can consider that 20 other topics around the topic have been noticed by the user.
2) If a topic has not been focused by a user after the topic was noticed by the user 3 or more than 3 times, we can consider that the user is not interested in the topic. That topic will be put into the user-unfocused topic list, and the user-unfocused time is defined as the last time when the user noticed the topic.

Where, the 20 topics that can be noticed by the user consist of the following parts:

1) 10 topics that have an earlier last reply time than the user-focused topic
2) 10 topics that have a later last reply time than the user-focused topic and the last reply time is earlier than the user-focused time

Note: We only select topics that closest to the user-focused topic.

2.2 Building Training Data

Since user-focused topics and user-unfocused topics are required as training instances in the maximum entropy model training, we use all available topics in user-focused/unfocused topic list as the training data when training the model.

2.3 Building Resource Data

When calculating topic features in training data or unranked forum topic list, user-focused topics are required to obtain some of the influence factors' occurrence probabilities. Thus, we generally use all available topics in the user-focused topic list as the resource data. Only when calculating topic features for the user-focused instances in the training data, since the user-focused instance itself is in the user-focused topic list, we only use other topics in the user-focused topic list as the resource data, except for the instance itself.

Moreover, when calculating some influence factors in the topic feature, topics posted by the user are also required. In these cases, we also use all available user posted topics as the resource data. The influence factors that required user posted topics will be noted in the following chapters.

3. Definition of Topic Feature

When a user surfing a forum, topic feature will influence the user's decision-making as to whether or not to view a topic. Thus, the topic feature is different for each user, and it is not an intrinsic property of a topic. With the investigation and analysis of users' browsing habit, I have proposed 10 factors that will influence a user's decision-making, and we use the 10 factors' occurrence probabilities as the topic feature. A topic feature is defined as an n-tuple $x = (P_1, P_2, ..., P_n)$. Where $P_1, P_2, ..., P_n$ are the occurrence probabilities of influence factors. The description of $P_1, P_2, ..., P_n$ are as follows:

1) Occurrence Probabilities of Influence Factors Related to the Topic

P_1: The probability that the topic semantic feature is similar to the user's content preference

P_2: The probability that the post time of the topic satisfies the user's requirement of timeliness

P_3: The probability that the last update (last reply) time of topic satisfies the user's requirement of timeliness

2) Occurrence Probabilities of Influence Factors Related to the Topic Author

P_4: The probability that the topic author's content preference is similar to the user's content preference

P_5: The probability that topics posted by the topic author are always focused by the user

P_6: The probability that topics posted by the topic author are always focused by forum users (including all the forum users)

P_7: The probability that the topic author's registration time satisfies the user's requirement of registration length

P_8: The probability that the topic author always post original articles

3) Occurrence Probabilities of Influence Factors Related to the users Topic Replier

P_9: The probability that the replies count of the topic satisfies the user's requirement of replies count

P_{10}: The probability that the topic replier feature is similar to the user's replier preference

3.1 Calculation Methods of the Influence Factors' Occurrence Probabilities

Calculation methods of the influence factors' occurrence probabilities can be divided into three classes. The occurrence probabilities and their corresponding calculation methods are as follows:

Occurrence Probabilities	Corresponding Calculation Method
P_1, P_4, P_{10}	Calculate cosine of the angle using the vector space model, and then obtain the probability.
P_2, P_3, P_7, P_9	Use topics in the resource data as standards, and then obtain the probability using "counting method".
P_5, P_6, P_8	Obtain the probability using a relatively simple expression.

Table 1: Calculation Methods of the Influence Factors' Occurrence Probabilities

3.2 Calculation of P_1, P_4 and P_{10}

Note: P_1 refers to the probability that the topic semantic feature is similar to the user's content preference; P_4 refers to the probability that the topic author's content preference is similar to the user's content preference; P_{10} refers to the probability that the topic replier feature is similar to the user's replier preference.

3.2.1 Creating Topic Semantic Feature Vector

1) Building Word Segmentation Dictionary

Before extracting the semantic information from topics, we need to first segment the topic contents and the topic tittle into words. With the aim of handling the internet language in forum topics, we build the word segmentation dictionary based on the internet word frequency database from Sogou Labs. We choose to use IDF as the weight of word, and define $\mathrm{D_i}$ as the frequency (occurrence count in webpages) of the $\mathrm{i^{th}}$ word in the word frequency database. Then, the weight of the word can be expressed as [13]: $\mathbf{IDF_i = \ln \frac{MaxD_i}{D_i}}$, where $\mathrm{MaxD_i}$ is the maximum $\mathrm{D_i}$ in the word frequency database. Finally, we put the 150,000 word weights obtained from the calculation into the word segmentation dictionary.

2) Extracting Keywords

We choose to use the Chinese word segmentation tool from Hyland Information Technology with the word segmentation dictionary described above to segment the topic contents and the topic tittle. Then, keywords in the topic can be obtained and we use $\mathrm{TF/IDF}$ [13] as the weights of the keywords. We sort the keywords by $\mathrm{TF/IDF}$ value in descending order, and put the first 50% keywords in the topic contents into the keywords database of the topic. Since the tittle express more semantic information in the topic, we put the first 50% keywords in the topic tittle into the keywords database and replace all the $\mathrm{TF/IDF}$ values with 1. After keywords extraction, some noise words are removed from the keywords database manually.

3) Creating Topic Semantic Feature Vector

We use a multidimensional space vector to describe the semantic feature of a topic. Then we make each keyword in the keywords database corresponds to a dimension of the vector and set each dimension value to the keyword $\mathrm{TF/IDF}$. Synonym expansion and dimension reduction are efficient ways to improve the performance of the vector space model [14][15]. However, they will not be discussed here as they are out of the research area of this paper.

3.2.2 Creating Users' Content Preference Vector

A user's content preference can be defined as the user's preference of topic content semantics. We can obtain a user's content preference vector by accumulating the semantic feature vectors of the topics in the user's resource data (including user posted topics). With the increase of resource data, user's content preference vector will get closer and closer to the user's real topic content preference.

3.2.3 Creating Topic Replier Feature Vector

We use a multidimensional space vector to describe the replier feature of a topic. Then we make each replier of the topic corresponds to a dimension of the vector and set the dimension values to 1. When comparing the topic replier feature vector with a user's replier preference vector, the corresponding dimension is needed to be removed from the topic replier feature vector if the user him/herself is a replier of the topic.

3.2.4 Creating Users' Replier Preference Vector

A user's replier preference can be defined as the user's preference of topic replier features. We can obtain a user's replier preference vector by accumulating the replier feature vectors of the topics in the user's resource data (including user posted topics). The corresponding dimension is needed to be removed from the user's replier preference vector if the user him/herself is a replier in the vector. With the increase of resource data, user's replier preference vector will get closer and closer to the user's real replier preference.

3.2.5 Vector Space Model and Vector Dimension Selection

In order to calculate P_1, P_4 and P_{10}, we need to make the following comparison between the vectors:

P_1: Comparison between the user's content preference vector and the topic semantic feature vector

P_4: Comparison between the user's content preference vector and the topic author's content preference vector

P_{10}: Comparison between the user's replier preference vector and the topic replier feature vector

Since the cosine similarity has been widely used and has a very good performance [16][17], we choose to use cosine similarity as the comparison method. Cosine similarity is designed to calculate the cosine of the angle between the vectors. Thus, the lengths of the vectors will not affect the compare results, so as to avoid the adverse effect on the compare results while comparing vectors that have a significant difference in length, especially when comparing a vector belonging to a user with a vector belonging to a topic.

Forum users usually have multiple interest areas. Different interest areas in the user's preference vector will affect each other when comparing the preference vector with other vectors. Specifically, if there are some keywords existed in the user's content preference vector but not existed in the topic semantic feature vector (or the topic author's content preference vector), keywords irrelevant to the topic (or the topic author) will have an adverse effect on the compare results. Similar problems are also existed when handling the user's replier preference vector. Moreover, when a user begin to be interested in a new area, since the dimension values for the user's new interest area are much lower than those for the existing interest areas, new dimensions in the user's preference vector will only have a very limited influence on the compare results.

In order to handle users' multiple interest areas, clustering has been widely used in the previous works. Articles are merged into clusters and the semantic feature vectors of the clusters are used as user's content preference [3][5][7][9][17][18]. As regards the new interest areas, forgetting algorithm has also been used to delete user's existing preferences regularly [7][18]. However, these techniques are low in speed and can not fully resolve the problems. For example, clustering method can not handle the keywords belonging to more than one area, and the forgetting algorithm may cause loss of some areas that the user is still interested in. In addition, there is no previously published solution to the problem of user's multiple replier preference in the published papers.

With the aim of resolving these problems, I have proposed a new technique called vector dimension selection, which has a much higher speed and does not have the shortcomings described above. When using the vector dimension selection, different interest areas will not affect each other in the comparison, and new interest areas are able to affect the compare results immediately. The approach of the vector dimension selection is described as follows:

1) When calculating the similarity between the user's content preference vector and the topic semantic feature vector, remove the dimensions existed in the user's content preference vector but not existed in the topic semantic feature vector.
2) When calculating the similarity between the user's content preference vector and the topic author's content preference vector, remove the dimensions existed in the user's content preference vector but not existed in the topic author's content preference vector.
3) When calculating the similarity between the user's replier preference vector and the topic replier feature vector, remove the dimensions existed in the user's replier preference vector but not existed in the topic replier feature vector.

After vector dimension selection, the cosine similarity between the vectors can be calculated with the following expression:

$$\cos\theta = \frac{\sum_{k=1}^{m} x_k y_k}{\sqrt{\sum_{k=1}^{m} {x_k}^2}\sqrt{\sum_{k=1}^{m} {y_k}^2}} \qquad \cos\theta \in [0, 1]$$

Where, m is the count of dimensions.

According to the research by Dong-Bo Zhan [16], the probability that two articles are similar increases linearly and monotonously with the increase of the cosine similarity between the semantic feature vectors of the two articles. Thus, the probability that two articles are similar can be approximately equal to the cosine similarity between the vectors. Furthermore, we can regard the comparison between the vectors of other types as the comparison between the semantic feature vectors. Therefore, we can obtain P_1, P_4 and P_{10} by calculating the cosine similarity between the vectors and use the cosine similarity as the value of P_1, P_4 or P_{10} approximately.

3.3 Calculation of P_2, P_3, P_7 and P_9

Note: P_2 refers to the probability that the post time of the topic satisfies the user's requirement of timeliness; P_3 refers to the probability that the last update (last reply) time of the topic satisfies the user's requirement of timeliness; P_7 refers to the probability that the topic author's registration time satisfies the user's requirement of registration length; P_9 refers to the probability that the replies count of the topic satisfies the user's requirement of replies count.

1) First, we define a topic condition as a set of the post time, last update time, author's registration time and replies count of a topic. In order to calculate the probability that a given topic condition satisfies the user's requirements. We need to establish the comparison standards of the user's requirements. If a topic has been focused by the user, it can be considered that the condition of the topic satisfies the user's requirements, and such a condition can be used as a standard in comparison. We use A_n, B_n, C_n and D_n ($n \geq 1$) to represent the topic conditions in the user's resource data, and define them as follows:

   A_n: The difference value between the user-focused time and the post time of the n^{th} topic in the user's resource data

   B_n: The difference value between the user-focused time and the last update time of the n^{th} topic in the user's resource data

   C_n: The difference value between the user-focused time and the author's registration time of the n^{th} topic in the user's resource data

   D_n: The replies count of the n^{th} topic in the user's resource data

2) We use A_0, B_0, C_0 and D_0 to represent the condition of the topic needed to calculate P_2, P_3, P_7 and P_9, and define them as follows:

   A_0: The difference value between the current time and the post time of the topic

   B_0: The difference value between the current time and the last update time of the topic

   C_0: The difference value between the current time and the author's registration time of the topic

   D_0: The replies count of the topic

   Note: When calculating A_0, B_0 and C_0 for topics in the unranked topic list, use the time when creating the personalized topic list as the current time, whereas when calculating for topics in the user's training data, use the user-focused time or user-unfocused time as the current time.

3) The calculation of P_2, P_3, P_7 and P_9 are based on the following assumption:
   a) If A_0 (or B_0) $\leq$ A_n (or B_n), it can be considered that the topic needed to calculate P_2 or P_3 satisfies the user's requirements of timeliness.
   b) If C_0 (or D_0) $\geq$ C_n (or D_n), it can be considered that the topic needed to calculate P_7 or P_9 satisfies the user's requirements of author's registration length or replies count.

Thus, we define the following expression:

$$\mathbf{P_2} = \frac{\textbf{The count of } \mathbf{A_0 \le A_n}}{\textbf{The count of the topics in the user's resource data}}$$

$$\mathbf{P_3} = \frac{\textbf{The count of } \mathbf{B_0 \le B_n}}{\textbf{The count of the topics in the user's resource data}}$$

$$\mathbf{P_7} = \frac{\textbf{The count of } \mathbf{C_0 \ge C_n}}{\textbf{The count of the topics in the user's resource data}}$$

$$\mathbf{P_9} = \frac{\textbf{The count of } \mathbf{D_0 \ge D_n}}{\textbf{The count of the topics in the user's resource data}}$$

Where, the value of n is 1 to m, respectively.

3.4 Calculation of P_5, P_6 and P_8

Note: P_5 refers to the probability that topics posted by the topic author are always focused by the user; P_6 refers to the probability that topics posted by the topic author are always focused by forum users (including all the forum users); P_8 refers to the probability that the topic author always post original articles.

"The author" mentioned below refers to the author of the topic needed to calculate P_5, P_6 or P_8. All the calculations described below are needed to be done with the data up to the user-focused time of the last topic in the user's resource data.

1) Calculation of P_5:

$$\mathbf{P_5} = \frac{\textbf{The count of the topics that is in the user's resource data and is posted by the author}}{\textbf{The count of the topics in the user's resource data}}$$

2) Calculation of P_6:

The calculation of P_6 is based on the following assumption: If a topic has been focused by all the forum users, it can be considered that the topic is always focused by forum users. Then, the probability that the topic is always focused by forum users can be expressed as:

$$\mathbf{P_6'} = \frac{\textbf{The focuses count of the topic}}{\textbf{The count of the forum users}}$$

Therefore, the value of P_6 can be considered as the average P_6' of the topics posted by the author, and the expression is defined as follows:

$$\mathbf{P_6} = \frac{\textbf{The summation of the focuses counts of the topics posted by the author}}{\textbf{The count of the forum users} \times \textbf{The count of the topics posted by the author}}$$

Note: The count of the forum users does not include the users that have not focused on any topic.

3) Calculation of P_8:

$$\mathbf{P_8} = \frac{\textbf{The count of the original topics posted by the author}}{\textbf{The count of the topics posted by the author}}$$

Note: The approach of determining the originality of a topic is as follows: Calculate the cosine similarities between the new topic and the existing topics, after a user has posted a topic. If all the cosine similarities are less than 0.95, the new topic can be considered as an original topic. Moreover, the semantic fingerprint can also be used to accelerate the comparison. However, it will not be discussed here as it is out of the research area of this paper.

3.5 Smoothing of the Probabilities

Sometimes, the actual probability may not be 0 when the calculation result of P_n equals to 0. This problem is due to data sparseness, and it is often occurs when the user's browsing history data is not

enough. Moreover, some errors will be caused in the following calculation if all the P_n equals to 0. Therefore, we need to replace the value of P_n with a non-zero value when P_n equals to 0. This approach is called smoothing technique. Some smoothing methods have been proposed previously, such as absolute discounting and deleted interpolation [19]. However, most of these methods are quite complex and designed for natural language processing. Thus, I have proposed a simple smoothing method, and the orderliness of the probabilities can be remained. After calculating P_n, replace the value of P_n using the following expression:

$$\mathbf{P_n} = \begin{cases} \mathbf{0.1} & \mathbf{P_n = 0} \\ \mathbf{P_n \times 0.9 + 0.1} & \mathbf{P_n > 0} \end{cases} \qquad \mathbf{n} \in \{\mathbf{1}, \mathbf{2}, \dots, \mathbf{10}\}$$

4. Maximum Entropy Model and Personalized Ranking

The personalized forum topic ranking system is based on the prediction of the click probability for each topic. Therefore, we need to integrate the factors that will influence the user's decision-making as to whether or not to view a topic. The conditional maximum entropy model can integrate all the factors affecting on the target event, without taking into account the structure of the factors. Weights of the factors can be obtained throughout the training. After training, the occurrence probability of the target event under the influence of all the factors can be acquired.

We choose to use the conditional maximum entropy model to obtain the click probability under a given topic feature, namely, to calculate $P(\text{Focused}|\overline{x})$. Where, $\overline{x}$ is the given topic feature.

4.1 Principle of the Maximum Entropy Model

The principle of the maximum entropy model is to build a model that satisfies all the observed data while making no assumptions, namely, keeping the maximum uncertainty. Specifically, it is to seek a model that has the maximum entropy probability distribution, while satisfies all the observed data. Therefore, the indicator functions are needed to be introduced to represent the properties of each instance in the observed data or the instance for the unranked topic. In this paper, we use 10 indicator functions and make each indicator function corresponds to an influence factor's occurrence probability. The indicator function is denoted by $f_i(x, y)$, where i is the ordinal number of the influence factor, x is the topic feature, y denote whether the topic is focused by the user, and $y \in \{\text{Focused}, \text{Unfocused}\}$. Then, we define $f_i(x, y)$ as follows:

$$\mathbf{f_i(x, y)} = \begin{cases} \mathbf{P_i} & \mathbf{(y = Focused) \wedge (P_i \neq 0)} \\ \mathbf{0} & \mathbf{otherwise} \end{cases} \qquad \mathbf{i} \in \{\mathbf{1}, \mathbf{2}, \dots, \mathbf{10}\}$$

The target of the maximum entropy model is to obtain the model $P(y|x)$. For each i, the expected value of f_i in the model $P(y|x)P(x)$ should equal to the expected value of f_i in the observed distribution $\overline{P}(x, y)$. This is called the constraint, which can be expressed as the following form:

$$\mathbf{E_P(f_i)} = \mathbf{E_{\overline{P}}(f_i)} \qquad \mathbf{i} \in \{\mathbf{1}, \mathbf{2}, \dots, \mathbf{10}\}$$

Where,

$$\mathbf{E_P(f_i)} = \sum_{\mathbf{x,y}} \mathbf{P(x)P(y|x)\, f_i(x, y)} \approx \sum_{\mathbf{x,y}} \mathbf{\overline{P}(x)P(y|x)\, f_i(x, y)}$$

$$\mathbf{E_{\overline{P}}(f_i)} = \sum_{\mathbf{x,y}} \mathbf{\overline{P}(x, y)\, f_i(x, y)}$$

The target model $P(y|x)$ should satisfy the constraint described above and has the maximum entropy distribution. The conditional entropy of $P(y|x)$ can be expressed as follows:

$$\mathbf{H(P)} = -\sum_{\mathbf{x,y}} \overline{\mathbf{P}}(\mathbf{x})\mathbf{P}(\mathbf{y}|\mathbf{x}) \log \mathbf{P}(\mathbf{y}|\mathbf{x})$$

4.2 Introduction of the Lagrange Multiplier

According the description above, seeking the maximum entropy model $P(y|x)$ is a constrained optimization problem:

$$\mathbf{p}^* = \underset{\mathbf{p}\in\mathbf{C}}{\mathbf{argmax}}\, \mathbf{H(p)} \qquad \mathbf{C} = \{\mathbf{p} \in \mathbf{P}|\mathbf{E_P}(\mathbf{f_i}) = \mathbf{E_{\overline{P}}}(\mathbf{f_i}), \mathbf{i} \in \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \dots, \mathbf{10}\}$$

Where, p^* is the target model $P(y|x)$. Then we need to introduce a weight λ_i (Lagrange multiplier) for each f_i, so as to convert the constrained optimization problem into an unconstrained optimization problem. Therefore, we can get the following expression:

$$\mathbf{P}(\mathbf{y}|\overline{\mathbf{x}}) = \frac{\mathbf{exp}\sum_{\mathbf{i}} \boldsymbol{\lambda}_{\mathbf{i}}\mathbf{f_i}(\overline{\mathbf{x}}, \mathbf{y})}{\sum_{\mathbf{y}'} \mathbf{exp}\sum_{\mathbf{i}} \boldsymbol{\lambda}_{\mathbf{i}}\mathbf{f_i}(\overline{\mathbf{x}}, \mathbf{y}')}$$

4.3 Training Maximum Entropy Model

The solutions of λs can be obtained by iterations, and the process is called the maximum entropy model training. Iteration algorithms such as GIS, IIS and SCGIS are commonly used in the previous works [10]. We choose to use the SCGIS algorithm [20] as the training method, since it has a faster convergence rate than other algorithms. According to the condition of our model, we have made the following optimization to the SCGIS algorithm:

1) In the SCGIS algorithm, slowing factor $\frac{1}{\max_{j,y} f_i(\overline{x}_j, y)}$ is introduced to avoid excessive updates to the λ_is. However, in our model, $f_i(x, y) \leq 1$. Therefore, $\max_{j,y} f_i(\overline{x}_j, y) = 1$, and we do not need the slowing factor.
2) The SCGIS algorithm need to loops over all the outputs, y. However, in our model, $f_i(\overline{x}, \text{Unfocused})$ is constant equal to 0. Thus, $\exp \sum_i \lambda_i f_i(\overline{x}, \text{Unfocused})$ is constant equal to 1, and we can simply use 1 instead of calculating the value of $\exp \sum_i \lambda_i f_i(\overline{x}, \text{Unfocused})$. Therefore, we only need to calculate when y equals to Focused, and we do not need the loop.
3) In the SCGIS algorithm, a two-dimensional array $s[j, y]$ is used to hold $\sum_i \lambda_i f_i(\overline{x}_j, y)$, and a one-dimensional array $z[j]$ is used to hold $\sum_y \exp(s[j, y])$. According to the discussion in above, we can only use a one-dimensional array $s[j]$ to hold the values when y equals to Focused, and $z[j]$ can be replaced with $\exp(s[j]) + 1$.

The following description of the SCGIS algorithm is based on the Goodman's version [20] and has optimized according to the previous discussion. Moreover, we have corrected a serious mistake in the Goodman's description.

```
//N is count of the training instances (including user-focused and user-unfocused)
s[1..N]=0, observed[1..10]=0
for i = 1 to 10
    for j = 1 to M    //f_i(x̄_j, y_j) corresponds to the focused instance when j≤M
observed[i]+= f_i(x̄_j, Focused)
```

Figure 2: Initialization of the SCGIS Algorithm

```
for i= 1 to 10
    expected=0
    for j = 1 to N
        expected+=f_i(x̄_j, Focused)*exp(s[j])/(exp(s[j])+1)
    δ_i=log(observed[i]/expected)
    λ_i+=δ_i
    for j = 1 to N
        s[j]+=f_i(x̄_j, Focused)*δ_i
```

Figure 3: One Iteration of the SCGIS Algorithm

With the iterating of the program in Figure 3, λs will gradually converge to certain values, and $|\delta|$s will gradually get smaller. We stop the iterative process when $|\delta_i| < E$ for each i, and E is defined as the iterative endpoint. In order to minimize the performance loss, we check the endpoint every 100 iterations. The selection of the iterative endpoint will be discussed in the following chapters.

4.4 Calculation of $P(\text{Focused}|\bar{x})$ and Personalized Ranking

According to the expressions in chapter 4.2 and the λs discussed in chapter 4.3, we can obtain the value of $P(\text{Focused}|\bar{x})$, where $\bar{x}$ is the topic feature of the unranked topic. Moreover, according to the discussion in chapter 4.3, we can replace $\sum_y \exp \sum_i \lambda_i f_i(\bar{x}, y')$ with $\exp \sum_i \lambda_i f_i(\bar{x}, \text{Focused}) + 1$. Thus, we can get the following expression:

$$P(\text{Focused}|\bar{x}) = \frac{\exp \sum_i \lambda_i f_i(\bar{x}, \text{Focused})}{\exp \sum_i \lambda_i f_i(\bar{x}, \text{Focused}) + 1}$$

When a user browsing the forum, the latest n topics will be regarded as unranked topics. In order to reduce the influence on the precision of the ranking system, n should be as large as possible, and the update frequency of the forum also need to be considered when determine the value of n. Typically, n should be greater than 100. Subsequently, we need to calculate the conditional probability $P(\text{Focused}|\bar{x})$ for the unranked topics. λs can be obtained by training the maximum entropy model with the latest training data regularly. User-focused and user-unfocused topics in the training data can be used as training instances $(\bar{x}, \text{Focused})$ and $(\bar{x}, \text{Unfocused})$, respectively. Finally, we can sort the unranked topics by the value of $P(\text{Focused}|\bar{x})$ in descending order and provide the personalized forum topic list to the user.

4.5 Accelerating the Training Speed

Since the tremendously slow training speed has greatly limited the practical applications of the maximum entropy model, many innovative training algorithms have been described in order to improve the training speed. However, the training speed is still too low to meet the requirements of practical applications. In this paper, I have proposed two methods to accelerate the training speed: 1. Iterative endpoint selection; 2. Indicator function normalization. According to previously published papers, these two methods have not been introduced to improve the training speed.

4.5.1 Iterative Endpoint Selection

According to the discussion in chapter 4.3, the maximum entropy model is trained with the optimized SCGIS algorithm, and the iterative endpoint E significantly determines the training speed. Since we only need to compare the values of the conditional probabilities for the topics and do not need the

exact values, we can accelerate the training speed with incomplete iteration. When the iterative process is incomplete, although the conditional probabilities obtained are not accurate, the orderliness of the predicted probabilities can be mostly remained. Thus, the precision of the personalized ranking system can be mostly remained too. The impact of the iterative endpoint on the ranking precision and training speed will be discussed in the experiment part.

4.5.2 Indicator Function Normalization

According to the discussion in chapter 4.3, a slowing factor is used in the SCGIS algorithm to avoid excessive updates to the λs. Although we don't use it in our training method since $f_i(x, y) \leq 1$, the slowing factor is implicitly equal to 1. In the SCGIS algorithm, the update value of each λ_i in one iteration is calculated through multiplying $\log(\text{observed}[i]/\text{ expected})$ by the slowing factor (the slowing factor is not added to our algorithm description). In the personalized system, since some influence factors in the topic feature often have very low occurrence probabilities (such as P_5 and P_6), their corresponding $f_i(x, y)$s are also significantly smaller than 1. Therefore, these indicator functions should have used a smaller slowing factor. When the slowing factor is too much larger than the values of these indicator functions, updates to the corresponding λs will be much smaller than what they could be. Thus, these corresponding λs will need more iteration counts than other λs to reach convergence. The global training speed will also get slow due to the low convergence rates of these λs.

According to the discussion above, I have proposed using the indicator function normalization to improve the training speed. The target of the indicator function normalization is to linearly adjust the instance values of all the indicator functions to the interval $[0.1, 1]$ (the minimum occurrence probability after smoothing is 0.1). The procedures of the indicator function normalization are as follows:

1) Selection of the Start Point and End Point:

The instance set of an indicator function is mainly distributed in a certain distribution interval. However, a few instance values in the instance set are distributed far away from the others. If we use the maximum and minimum value of instance values as the start and end point, the interval range will be too large and the performance of the indicator function normalization will be limited. Therefore, we use the following rule to determine the distribution interval of the instance set of an indicator function: The distribution interval needs to contain 90% of the instance values. The specific procedures are as follows:

a) Sort the instance values in ascending order.

b) Use the maximum value of the first 5% instance values as the start point of the distribution interval, and the start point is denoted by SP.

c) Use the minimum value of the last 5% instance values as the end point of the distribution interval, and the end point is denoted by EP.

2) Linear Adjustment of the Instance Values

The linear adjustments of the instance values need to conform to the following rules:

a) Linearly expand (or contract) the instance values that in the distribution interval to the middle 90% of the interval $[0.1, 1]$, namely, $(0.145, 0.955)$.

b) Linearly expand (or contract) the instance values that are smaller than SP to the first 5% of the interval $[0.1, 1]$, namely, $[0.1, 0.145]$.

c) Linearly expand (or contract) the instance values that are larger than EP to the last 5% of the interval $[0.1, 1]$, namely, $[0.955, 1]$.

According to the rules above, the indicator function normalization can be expressed as follows:

$$\mathbf{f(x)} = \begin{cases} \mathbf{0.1} & \mathbf{x = SP = 0.1} \\ \dfrac{\mathbf{0.045(x - 0.1)}}{\mathbf{SP\ -\ 0.1}} + \mathbf{0.1} & \mathbf{x \le SP, SP \neq 0.1} \\ \mathbf{1} & \mathbf{x = EP = 1} \\ \mathbf{1 - \dfrac{0.045(1 - x)}{1 - EP}} & \mathbf{x \ge EP, EP \neq 1} \\ \mathbf{0.55} & \mathbf{x = EP = SP} \\ \dfrac{\mathbf{0.81(x - SP)}}{\mathbf{EP - SP}} + \mathbf{0.145} & \mathbf{SP < x < EP, EP \neq SP} \end{cases}$$

Where, x is the instance value of the indicator function. The impact of the indicator function normalization on the ranking precision and training speed will be discussed in the experiment part.

5. Experiments

The experiments were done in offline mode. We have calculated the precision of the personalized ranking system based on the historical data of the forum database. The experiments are divided into two parts: 1. Compare of the precision between the personalized ranking system and the traditional ranking scheme, and evaluate the impacts of the vector dimension selection and the user's browsing history amount; 2. Evaluate the impacts of the training optimization methods on the ranking precision and training speed.

Moreover, in the experiments, we did not use the influence factor corresponding to P_8, since most of the topics in our database are discussions between the students and most of them are original.

5.1 Experimental Database and Running Environment

The experimental database comes from the forum of Beijing No.8 Middle School, which was existed from April 2003 to July 2006 and managed by me. There were 773 registered users and 2302 topics (not included the replies) in the forum. After excluded the users that had not focused on any topic, there were 221 users in the forum.

In the experiment, our applications were developed in C#, and running on Microsoft .NET Framework 3.5. The experimental database was running on Microsoft SQL Server 2008. The computer used in the experiments was equipped with Intel Core2 Q9300 (2.53 GHz) and 8 GB RAM. All the running time data in the experimental results is based on the system described above.

5.2 Building Resource Data, Training Data and Test Data

In order to build the resource data, training data and test data, we need to first create the user-focused/unfocused topic list according to the method described in chapter 2.1. We select 20 latest user-focused topics from the user-focused topic list and 20 latest user-unfocused topics from the user-unfocused topic list, respectively. Then, we use the 40 topics selected from the topic lists as the test data.

In order to evaluate the impact of the user's browsing history amount on the ranking precision, we need to select some topics from the rest of the user-focused/unfocused topic list, and use these

topics as the available user-focused/unfocused topic list. In the experiments, topics in the available user-focused/unfocused topic list are used to build the resource data and training data.

We select the earliest n topics from the rest of the user-focused topic list as the available user-focused topic list, and denote the count of the user-focused topics as Nn. Then, we use the user-focused time of the last topic in the available user-focused topic list as the reference time. Moreover, when the topics posted by the user is needed to calculate some of the influence factors, we will also select the topics that posted by the user and the post time is earlier the reference time as the resource data.

We select the topics whose user-unfocused time is earlier than the reference time from the rest of the user-unfocused topic list as the available user-unfocused topic list. The following figure is shown the division of the user-focused/unfocused topic list.

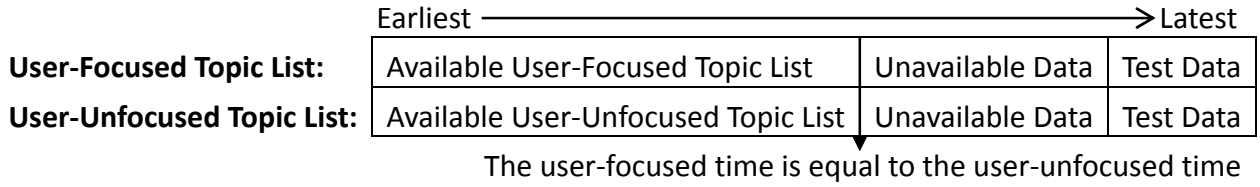


Figure 4: The Division of the User-Focused/Unfocused Topic List

Since the resource data and training data are built from the available user-focused/unfocused topic list and the user-unfocused topic list is built from the user-focused topic list, the amount of the resource data and training data is determined by the count of the user-focused topics. Thus, in the experiments, user's browsing history amount is expressed by Nn.

5.3 Evaluation Indicators of the Personalized Ranking System

After ranking the topics in the test data according to the method described in chapter 4.4, we used the precision of first 5/10/15/20 topics as indicators to evaluate the precision of the personalized ranking system, since there are 20 user-focused topics in the 40 test topics. The precision of first n topics can be calculated by: **n'/n**, where n' is the count of the topics that actually focused by the user in the first n topics.

In the experiments, we have also used the current ranking scheme, which ranks the topics by last update time, as a control ranking method. In the following chapters, the current ranking scheme will be called as "traditional ranking". The calculation method of the evaluation indicators of the traditional ranking is similar to that for the personalized ranking system.

5.4 Precision Evaluation of the Personalized Ranking System

5.4.1 Experimental Setting

In the precision evaluation, we have evaluated the impacts of the browsing history amount Nn and the vector dimension selection. Moreover, we have compared the results with the traditional ranking. We set the initial value of Nn to N10, and incremented n by 5. The experiments were stopped when Nn reached N60, since only 10 users could meet the requirements of N60.

As shown by a small-scale experiment, when we use 0.0005 as the iterative endpoint, the difference value between the predicted probability and the probability obtained from a completely converged model is less than 1%. Therefore, in all the precision evaluation, we used 0.0005 as the iterative endpoint and disabled the indicator function normalization, so as to ensure the model training is complete.

5.4.2 Experimental Results

Since the forum database is obtained from the current forum system, some users in the forum database have an "abnormal behavior", which is replying all the topics in the first page of the forum. This "abnormal behavior" is existed in the current ranking scheme, and used to achieve popularity. Since most of the topics replied by the user is existed in the first page, the precision of the traditional ranking is relatively high. Although the maximum entropy model is able to increase the weight of P_3 (related to the last update time) according to the user's behavior, the "abnormal behavior" is often irregular and not continuous, and sometimes even only presented in the test data. Thus, the precision of the personalized ranking system is low in the users with "abnormal behavior".

However, the "abnormal behavior" is only existed in the current ranking scheme. If forum systems build topic lists according the personalized ranking system, every user will get a unique topic list. Then the "abnormal behavior" will be meaningless and no longer existed. Therefore, we have excluded the users with "abnormal behavior" from the following statistics.

After excluded 6 users with "abnormal behavior", there were 26 users in the N10 group and 10 users in the N60 group. The average number of the users in groups was 16.

In the experiments, we have calculated the average precision of all the users for each group and its increment rate (compared to traditional ranking). The experimental results are shown in the following figures. The legend of the figures is defined as "enabled/disabled the vector dimension selection (-increment rate)". In the figures, "traditional ranking" is abbreviated to "TR".

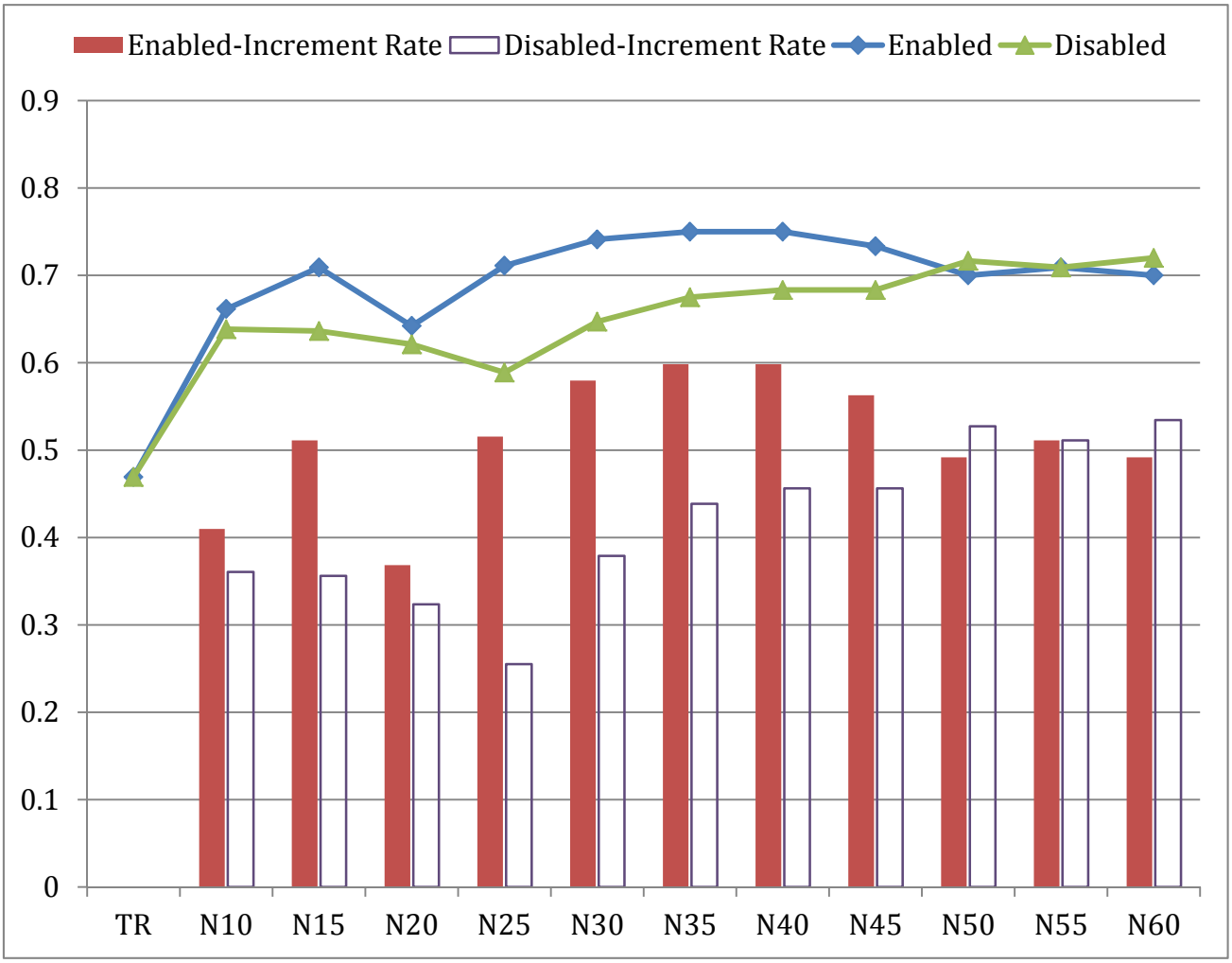


Figure 5: Average Precision of First 5 Topics and its Increment Rate (Compared to TR)

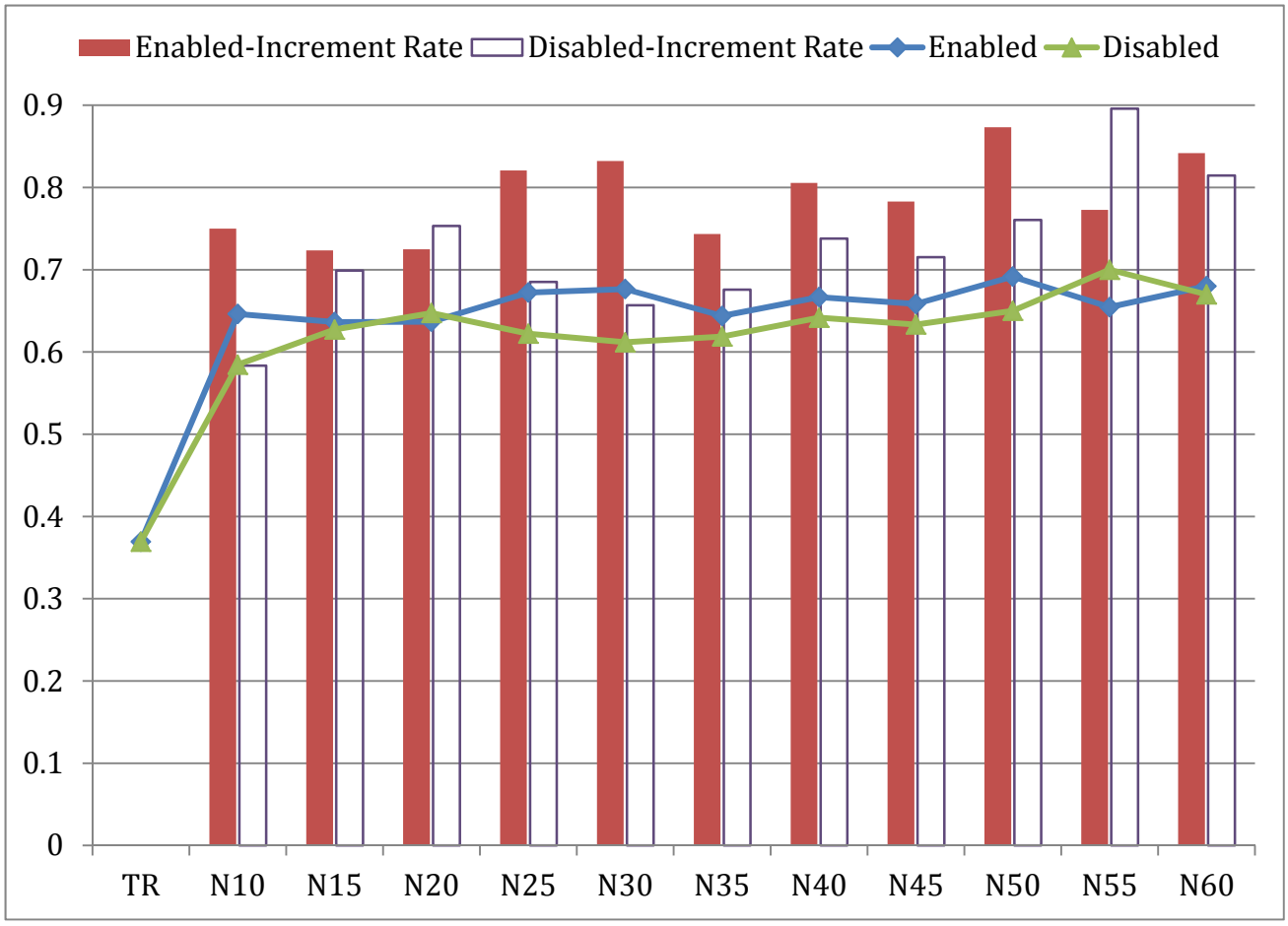


Figure 6: Average Precision of First 10 Topics and its Increment Rate (Compared to TR)

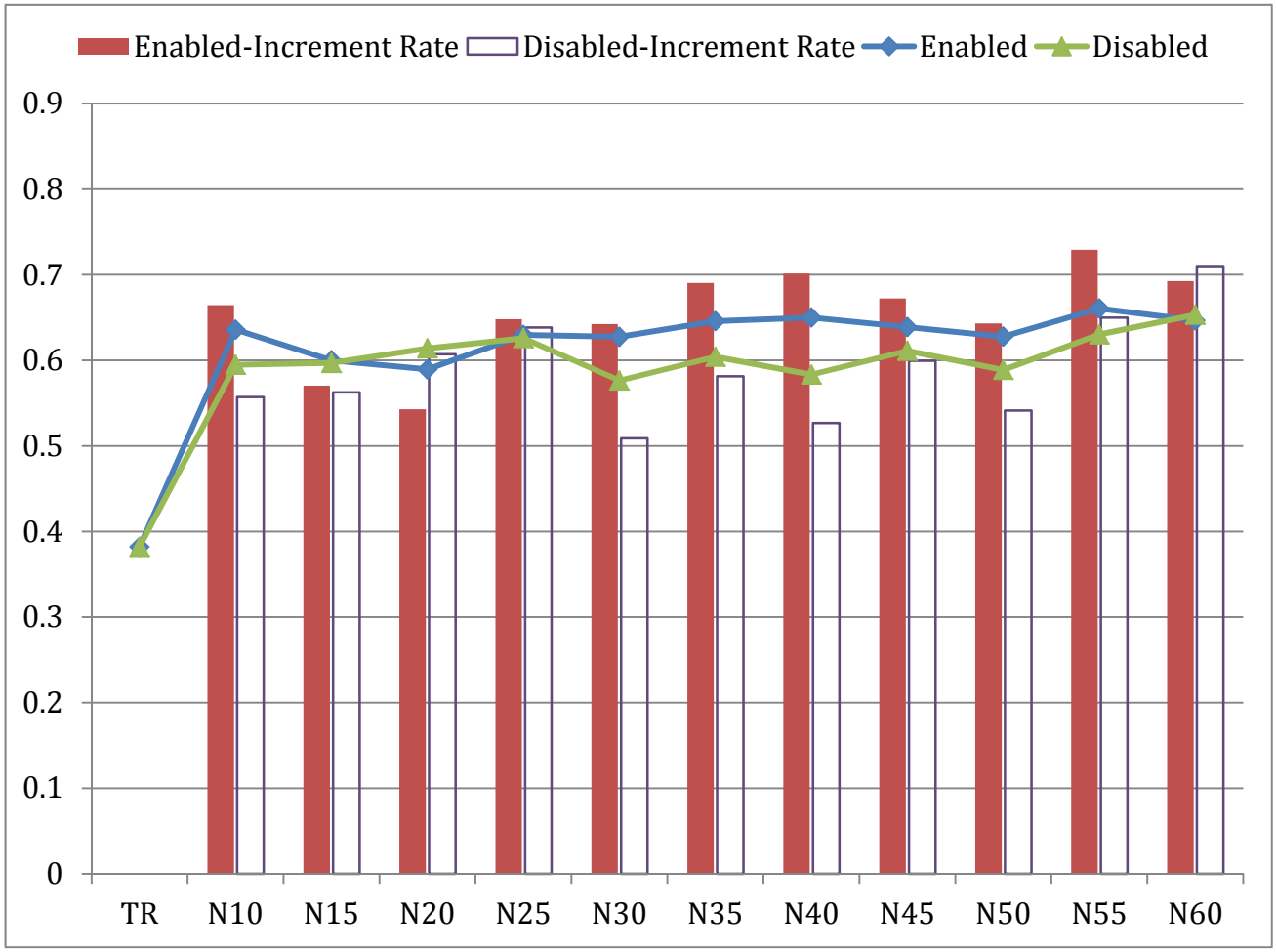


Figure 7: Average Precision of First 15 Topics and its Increment Rate (Compared to TR)

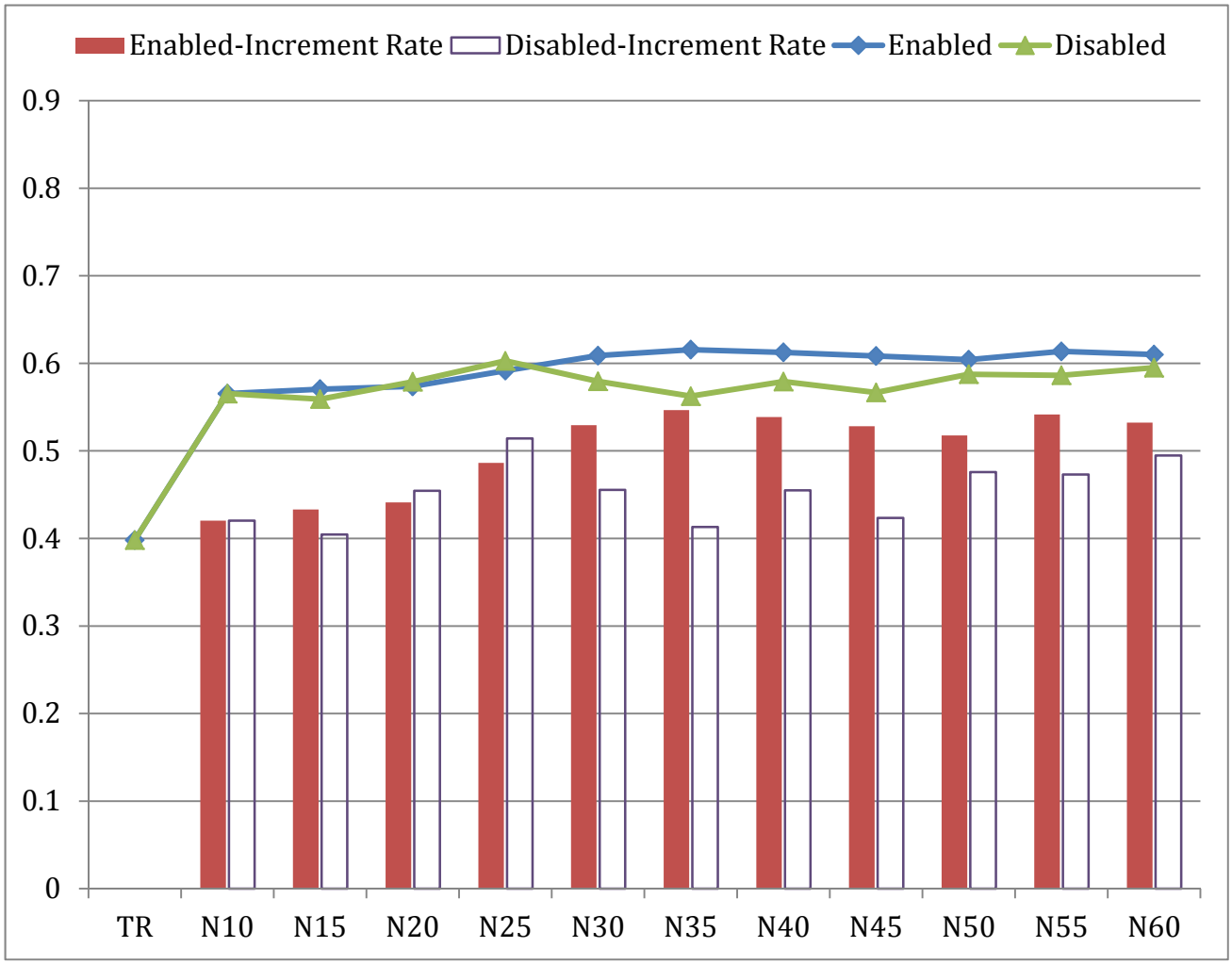


Figure 8: Average Precision of First 20 Topics and its Increment Rate (Compared to TR)

As shown in the figures above, the precision of the traditional ranking is about 37% to 47%. However, the precision of random ranking is 50% in theory. It indicates that the precision of the traditional ranking is quite low. Users in current forums are not able to find the topics they needed conveniently.

On the other hand, when a user has focused on 10 or more than 10 topics, the personalized ranking system will be able to provide personalized forum topic lists to the user. In all the experimental groups, the average precision of first 5/10/15/20 topics is significantly higher than that in the traditional ranking. Moreover, when a user has focused on more than 25 topics and the vector dimension selection is enabled, the precision of the personalized ranking system is about 60% to 75%, which improves the traditional ranking by 50% to 85%. It indicates that such a personalized ranking system has significantly improved users' efficiency when obtaining information from forums. With the increase of user-focused topics, the precision of the personalized ranking system will slightly increase or generally remain stable.

After enabled the vector dimension selection, the precision of the personalized ranking system is generally increased. It indicates that the vector dimension selection has a very good performance in handling users' multiple interest areas and new interest areas.

Furthermore, since the database used in the experiments was obtained from the current forum system, topics with later last update time have acquired more opportunities to be browsed by the users. Therefore, users may have missed some topics that they are interested in, which makes the experiments inequitable to the personalized ranking system. Thus, in practice, the personalized

ranking system will have a better performance than the experimental results, whereas the traditional ranking will have a lower precision than the experimental results.

5.5 Evaluation of the Training Optimization Methods

5.5.1 Experimental Setting

In the experiments, we have evaluated the impacts of the iterative endpoint and the indicator function normalization. We set the initial value of Nn to N10, and incremented n by 5. The experiments were stopped when Nn reached N60.

As shown by a small-scale experiment, when we use 0.0005 as the iterative endpoint, the difference value between the predicted probability and the probability obtained from a completely converged model is less than 1%. Therefore, we use 0.0005 as the baseline of the iterative endpoint. In the experiments, when we use 0.02 as the iterative endpoint, most of the users will only need 100 iterations (we check the endpoint every 100 iterations). Thus, we choose 0.02 as the end point of the iterative endpoint.

In order to avoid the influence of the vector dimension selection, we have disabled the vector dimension selection in all the experimental groups.

5.5.2 Experimental Results

In the experiments, we have calculated the average precision of first 10 topics of all the users for each group. The experimental results are shown in the following figures. The legend of the figures is defined as “iterative endpoint-enabled/disabled the indicator function normalization”.

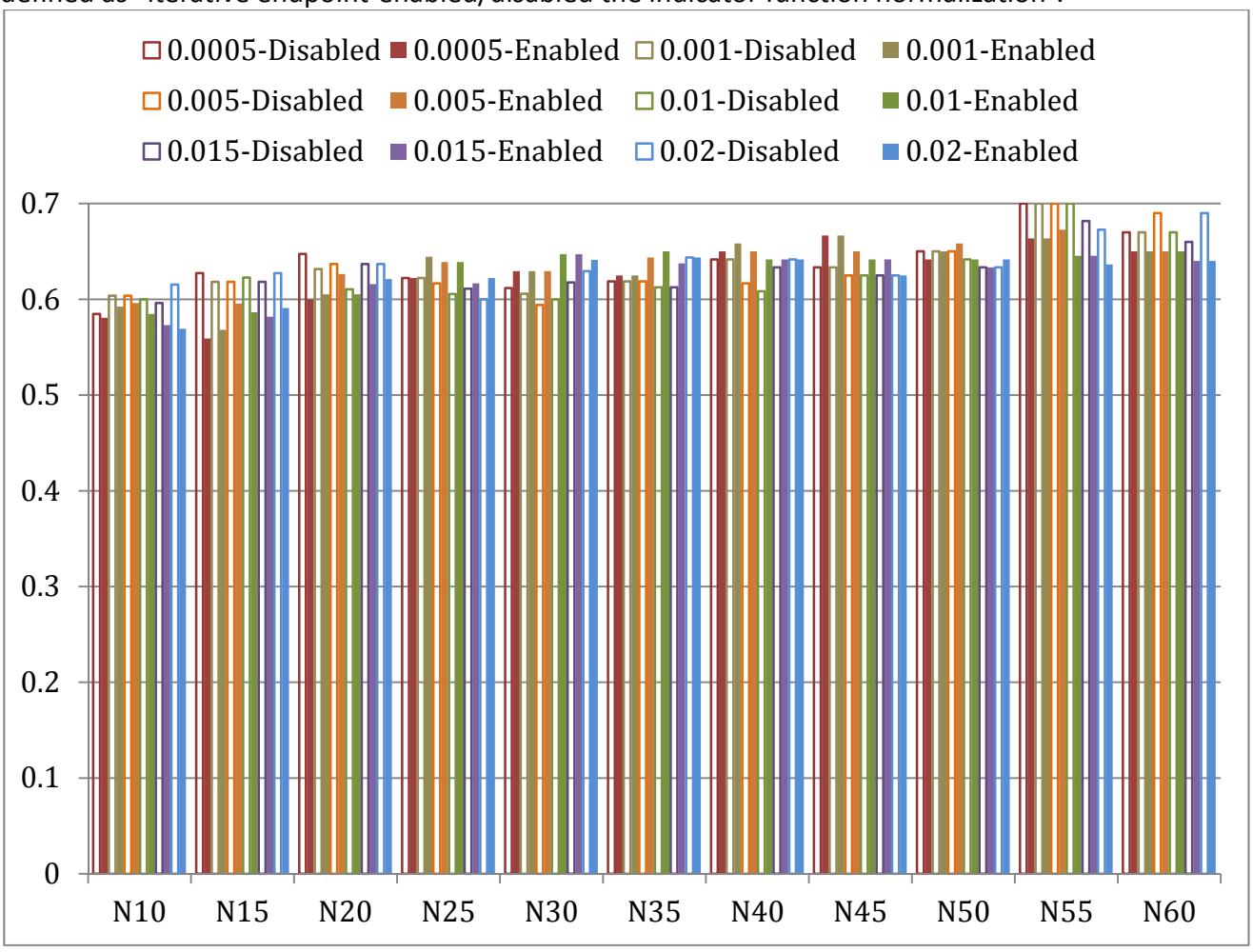


Figure 9: Average Precision of First 10 Topics

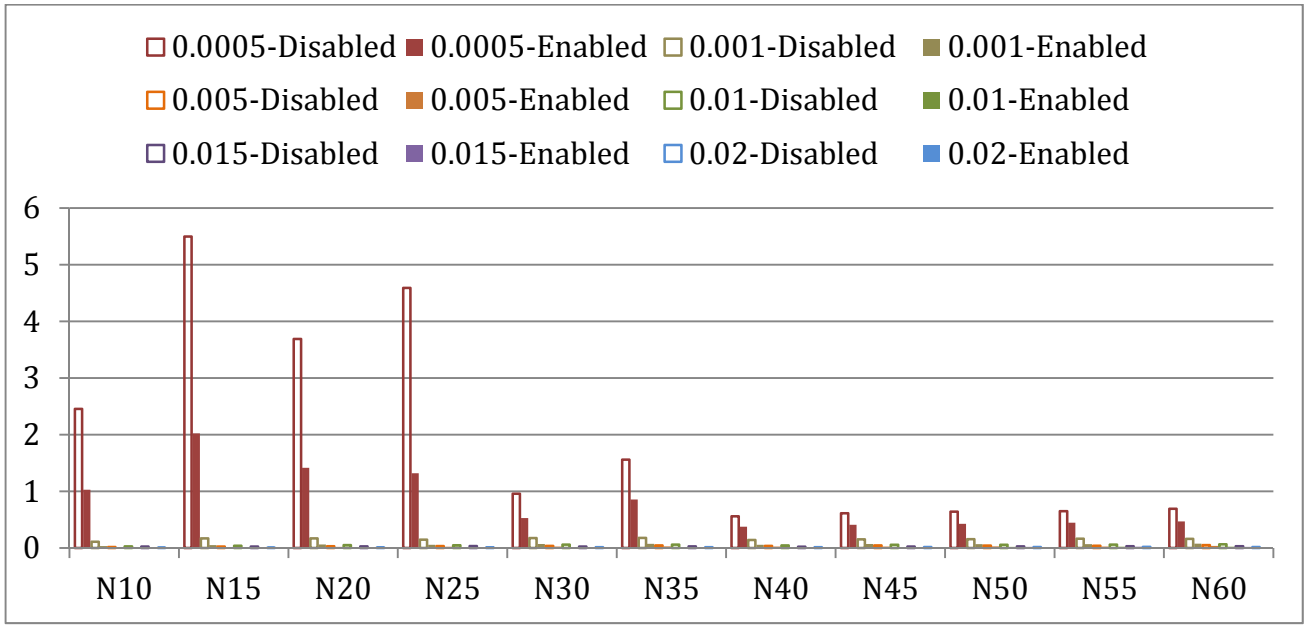


Figure 10: Average Training Time of the Maximum Entropy Model (Seconds)

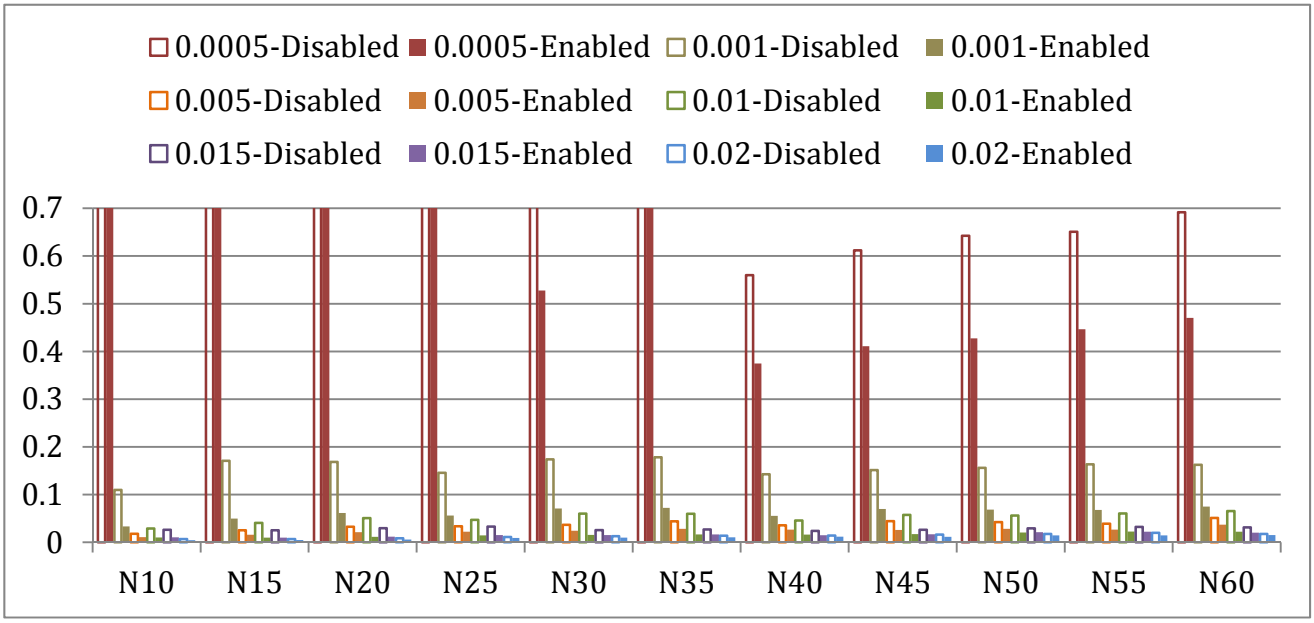


Figure 11: Magnifying Part of Figure 10 (0s to 0.7s)

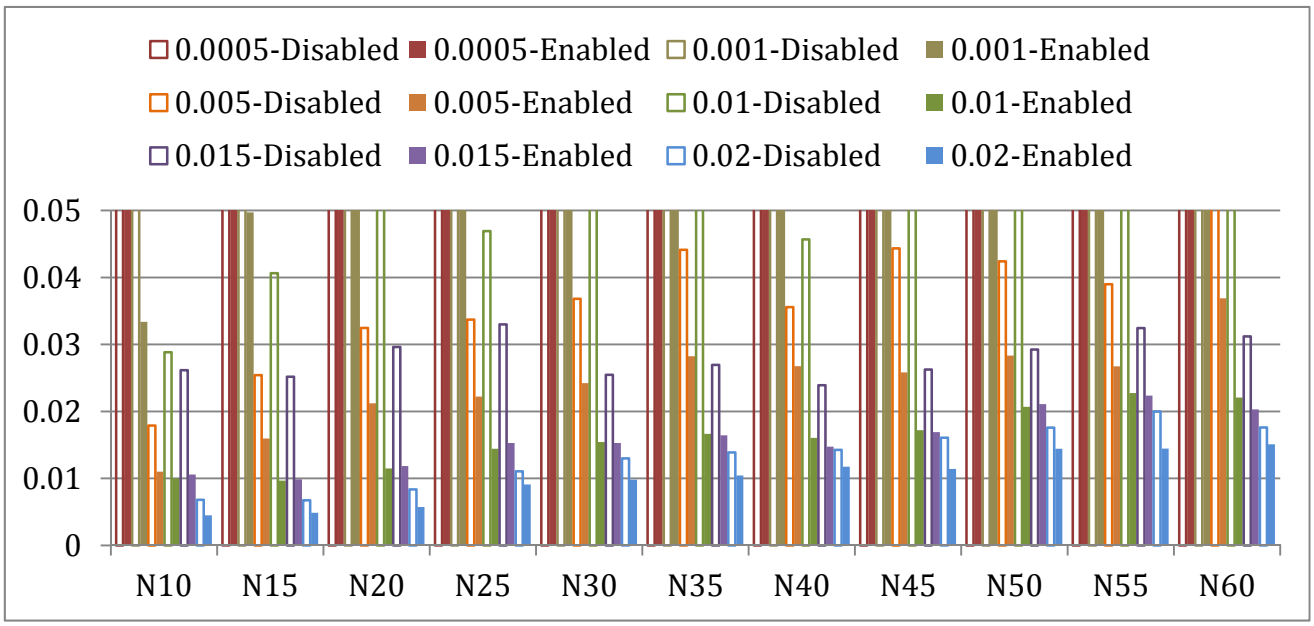


Figure 12: Magnifying Part of Figure 10 (0s to 0.05s)

As shown in the figures above, the iterative endpoint has little or no influence on the precision of the personalized ranking system. When we use 0.02 as the iterative endpoint, the model training is very incomplete (tens of thousands of iterations are needed to reach endpoint 0.0005, whereas only hundreds of iterations are needed to reach endpoint 0.02). Since we only need to compare the values of the conditional probabilities for the topics and do not need the exact values, the precision of the personalized system is generally stable. On the other hand, with the increase of the iterative endpoint, the training time has greatly decreased. When we use 0.02 as the iterative endpoint, the training time of all groups are less than 0.02 seconds, which makes the training time no longer be the bottleneck of the maximum entropy model.

After enabled the indicator function normalization, the training time in all the experimental groups has decreased by about 50%. However, the precision of the personalized ranking system is generally stable or sometimes even slightly higher. If only very limited training time is allowed in the running environment, we can use both of the iterative endpoint selection and the indicator function normalization to achieve a higher speed improvement, since the indicator function normalization is able to reduce the iteration counts that are needed to reach a iterative endpoint. When we use 0.02 as the iterative endpoint and using the indicator function normalization, the training time can be lowered to an average of 0.01 seconds for each user

6. Conclusion and Discussion

We have proposed a personalized forum topic ranking system based on the maximum entropy model. As shown by the experiments, when the user has focused on more than 25 topics and the vector dimension selection is enabled, the precision of the personalized ranking system is about 60% to 75%, which improves the traditional ranking by 50% to 85%. It indicates that such a personalized ranking system has significantly improved the quality of the topic list. Therefore, users are able to find their favorite topics easier and have a better efficiency when obtaining information from forums.

According to previously published papers, there is no existing research on personalized forum topic browsing system or maximum entropy model based personalized browsing system. Thus, we can only make comparisons with the existing personalized web page browsing systems [3][4][5] [8]. The precision of these systems with the optimum parameters are about 55% to 85%, which are similar to our forum topic ranking system.

In order to handle users' multiple interest areas and new interest areas, I have proposed using vector dimension selection to supersede the clustering method. The new method has a much higher speed than the clustering method. And in the experiments, the vector dimension selection has presented a very good performance.

In other research areas, the main shortcoming of the maximum entropy model is the tremendously slow training speed. Such a problem has greatly limited the practical applications of the maximum entropy model. However, we only use 10 factors in our personalized ranking system, while most of the natural language processing systems will use tens or even hundreds of thousands of factors. Moreover, with the normalization of the indicator functions and the selection of the iterative endpoint, the training speed has been significantly improved. When we use 0.02 as the iterative endpoint and using the indicator function normalization, the training time can be lowered to

an average of 0.01 seconds for each user. It indicates that such a model is able to meet the requirements of practical applications.

With the rapid increase of information, people need a more convenient way to obtain the information they need. People hope the computer can provide the information they want automatically. The personalized forum topic ranking system will give users a new way to find their favorite topics easily. Although the personalized ranking system is designed for forums, it can also be used in other areas after some modifications, such as personalized library and personalized news. It can also be used in personalized search, if we add some factors related to the similarity.

7. Research Prospect

7.1 Improvement of the Maximum Entropy Model Training

Although the maximum entropy model has a very high training speed, we are still trying to improve the training speed with the following ways:

1) **Training algorithm for Incremental Instances**

   A user's training instances are developing with his/her browsing process. The new instances also have some relationship to the existing instances. However, the current training algorithm need to perform a new training process if there are some updates in the training data, without using the existing results. We are planning to design a new training algorithm that is able to perform the new training process based on the existing training results, so as to further improve the training speed.

2) **Parallel Training algorithm**

   With the increasing of the number of CPU cores, parallel computing will play an important role in the future. We are planning to design a parallel training algorithm so as to utilize the advantages of the parallel computing.

7.2 Improvement of the Personalized System

1) **Forgetting Algorithm**

   The user's preference is changing all the time. Moreover, too much preference data will slow down the personalized ranking system. Thus, we need to design a forgetting algorithm and selectively delete some of the user's browsing history regularly.

2) **Feature Selection Algorithm**

   As shown in the experiments, some weights of the influence factors are very low in some users. Thus, calculating of these factors is a waste of the system resources. We can design a feature selection algorithm [10][22] and only calculate the factors that the user is needed.

8. Practical Application Plan

In practical application, the developer of the personalized forum system will establish a center server. All the forums using the personalized forum system will be regard as member forums. Users' browsing history that in the member forums will be sent to the center server, and the center server will record the browsing history for each user. The center server will train the maximum entropy model for each user regularly, and then send the factor weights and the resource data to every member forums. When a user surfing a member forum, the member forum will be able to provide personalized forum topic lists for the user, according to the data received from the center server.

Moreover, when a user begins to surf a new forum, the forum system will also be able to use the preference data collected in other member forums.

References

[1] China Internet Network Information Center, The 24th Statistical Report on Internet Development in China, 2009

[2] Xiao-Ming Li, Hong-Fei Yan and Ji-Min Wang, Search Engine - Principles, Technology and Systems, Science Press, 2004

[3] Hai-Jin Quan, The User Interests Model with Real-time Updated User Interests Based on User Behaviors and Similar Semantic, Master thesis, Southwest China Normal University, 2005

[4] Hua-Yue Chen and Zheng-Yu Zhu, Personalization Recommendation Based on User Current Interest View, Computer Engineering, Vol.31 No.20 Oct.2005

[5] Guan-You Fu, Mining Web User Interest Based on the Analysis of User Browser Behavior, Master thesis, Chongqing University, 2004

[6] Yu Zhang and Fang Yuan, A User Interest Model-Based Personalized Information Retrieval Method, Journal of Shandong University(Natural Science), Vol.41 No.3, 2006

[7] Hong-Xiao Fei, Yi Dai, Jun Mu, Qin-Jing Huang and Xin-Hua Xiao, Establishing and Updating of User Profile In Personalized Information Filtering System, Computer Systems Applications, 2007

[8] Jie-Feng Shen, The Personal Information Recommendation System Base on Users' Interest, Master thesis, Xihua University, 2006

[9] Zheng-Yu Zhu, Xiao-Lin Zhang, Qian Xiong and Qi-Hong Xie, An Algorithm of Collaborative Recommendation Based on User's Interest Sub-Class, Computer Science, Vol.32 No.10, 2005

[10] Ya-Qian Zhou, Maximum Entropy Method and Its Applications in Natrual Language Processing, PhD thesis, Fudan University, 2004

[11] Guang-Tao Si, Pei-Feng Li, Qiao-Ming Zhu and Jun-Hui Li, A Method of Mail Filtering Based on Maximum Entropy Model, Computer Applications and Software, Vol. 25 No.1 Jan.2008

[12] Rong-Lu Li, Jian-Hui Wang, Xiao-Peng Tao and Yun-Fa Hu, Using Maximum Entropy Model for Chinese Text Categorization, Journal of Computer Research and Development, 42(1): 94 ~ 101, 2005

[13] Jun Wu, How to Measure the Similarity Between Queries and Web Pages, Google Blog, 2006, http://www.googlechinablog.com/2006/06/blog-post_27.html

[14] Yuan-Chao Liu, Xiao-Long Wang, Zhi-Ming Xu and Yi Guan, A Survey of Document Clustering, Journal of Chinese Information Processing, 2006(03)-0055-08

[15] Jia-Ni Hu, Jun Guo, Wei-Hong Deng and Wei-Ran Xu, Independent Semantic Feature Extraction Algorithm based on Short Text, Journal on Communications, Vol.28 No.12 Dec. 2007

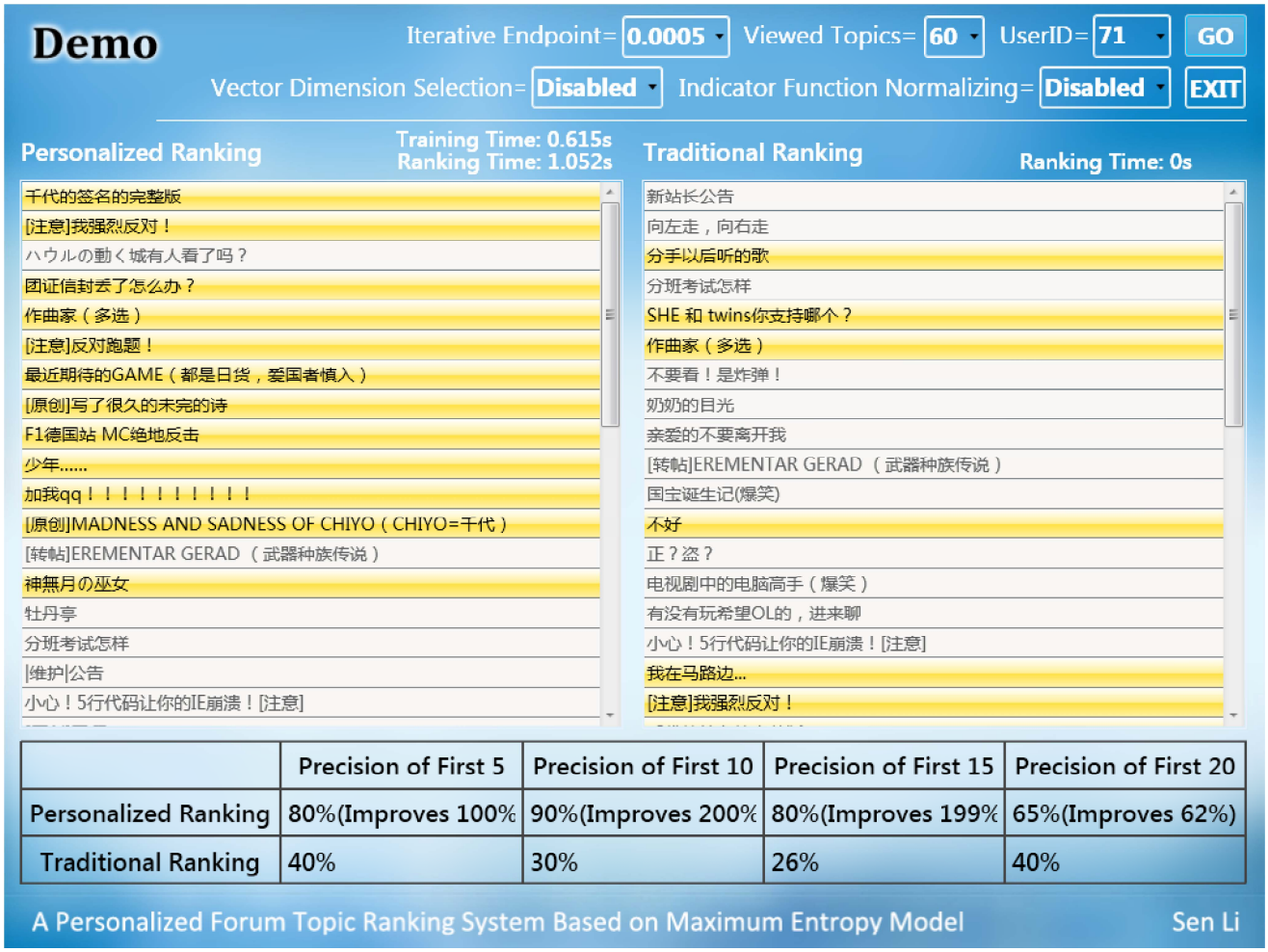
[16] Dong-Bo Bu, The Principles of Clustering/Classification and their Applications in Text Mining, PhD thesis, Chinese Academy of Sciences, 2000

[17] Yuan-Chao Liu, Xiao-Long Wang, Bing-Quan Liu and Bin-Bin Zhong, A Cluster-based Approach on Mining Text Preference, Application Research of Computers, 2005

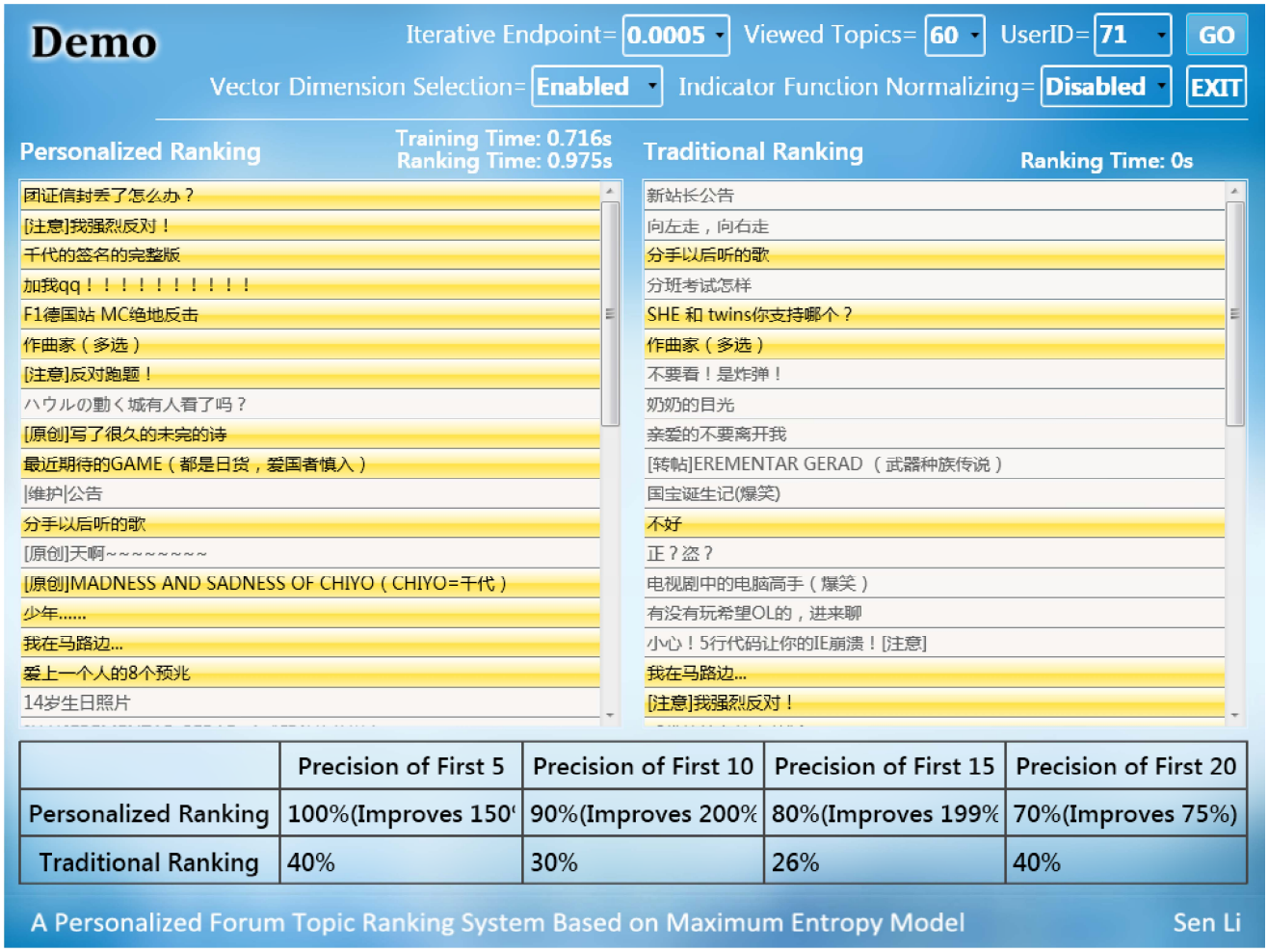
[18] Ping Jiang, The Research and Design on Personal Model Based on Mining User's Interests, Master thesis, Soochow University, 2005
[19] Rong-Lu Li, N-Gram and Smoothing Techniques
[20] Joshua Goodman, Sequential Conditional Generalized Iterative Scaling, Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002
[21] William Navidi, Statistics for Engineers and Scientists, Tsinghua University Press, 2007
[22] Yang-Sen Zhang, Yuan-Da Cao and Shi-Wen Yu, Improvement of Feature Selection Algorithm in Maximum Entropy Model and Disambiguation of Error-Correction Candidates, Transactions of Beijing Institute of Technology, Vol. 26 No. 1 Jan. 2006
[23] Jun Wu, The Law of Cosines and News Classification, Google Blog, 2006, http://www.googlechinablog.com/2006/07/12.html
[24] Jun Wu, Entropy - the Measurement of Information, Google Blog, 2006, http://googlechinablog.com/2006/04/4.html
[25] Jun Wu, Don't Put all of Your Eggs in One Basket - Maximum Entropy Principles, Google Blog, 2006, http://googlechinablog.com/2006/10/blog-post.html
[26] Jun Wu, The Extension of HMM and Bayesian Networks, Google Blog, 2007, http://googlechinablog.com/2007/01/bayesian-networks.html
[27] Thomas M. Cover and Joy A. Thomas, Elements of Information Theory (2nd Edition), China Machine Press, 2008
[28] Sheldon M. Ross, A First Course in Probability (7th Edition) , Posts and Telecom Press, 2007
[29] Marvin L. Bittinger, Calculus and Its Applications (8th Edition) , China Machine Press, 2006
[30] Jun Wu, Maximum Entropy Language Modeling with Non-Local Dependencies, 2002
[31] Xian-Tao Liao, Maximum Entropy Principle and its application, IR_Lab, 2005
[32] Laputa, Maximum Entropy Model and Natural Language Processing, NLP Group, AI Lab, Tsinghua Univ.
[33] 6.050J - Information and Entropy, MIT Open Course, http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-050JSpring-2008/CourseHome/index.htm

Appendix: Screenshots of the Demo

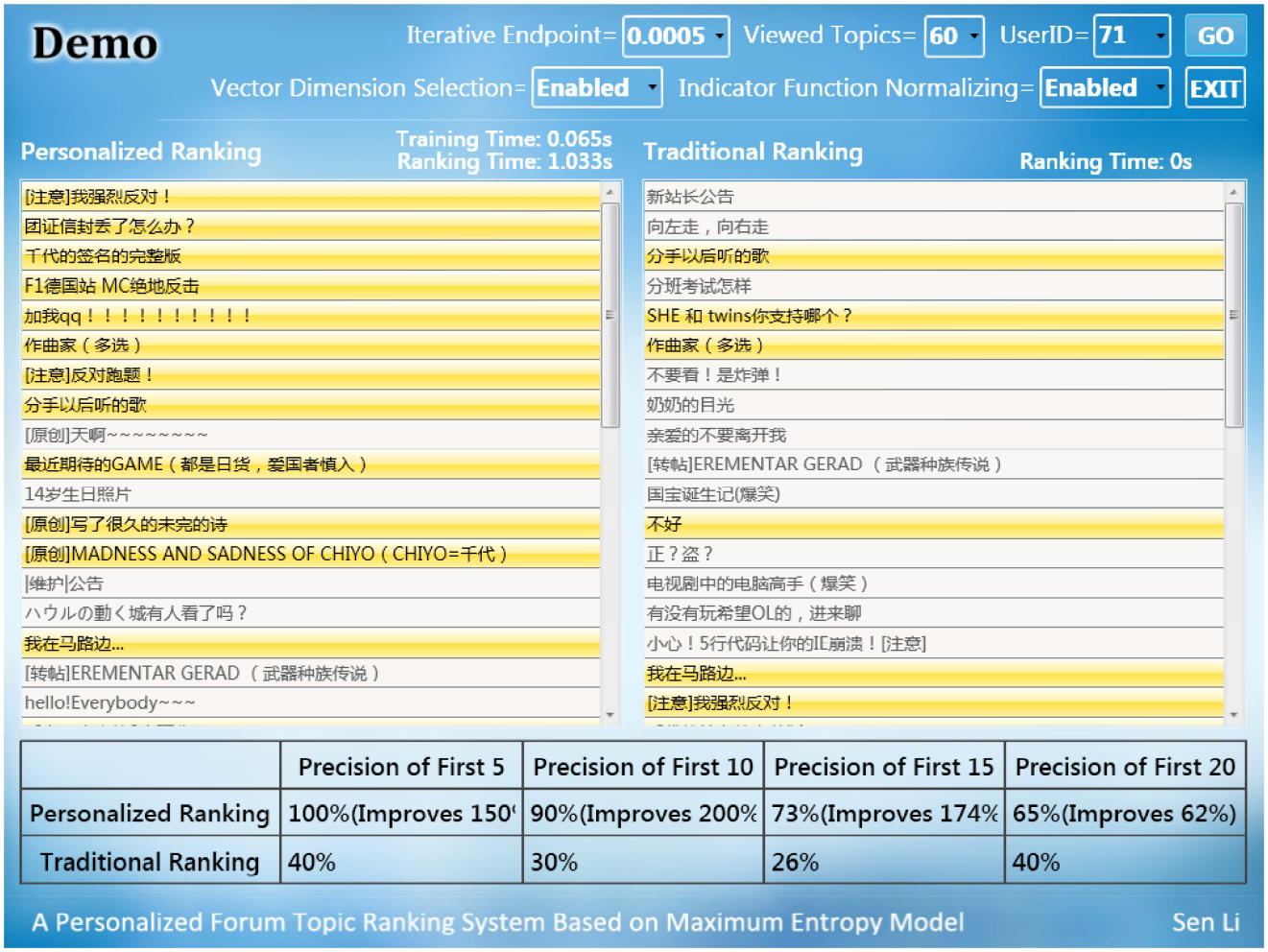
The following screenshots are a user's real browsing experience. The demo has made a comparison between the traditional ranking and the personalized ranking system. The yellow highlighted topics are the topics that actually focused by the user, namely, the topics that the user is interested in. The screenshots has also shown the impacts of the iterative endpoint, the indicator function normalization and the vector dimension selection.



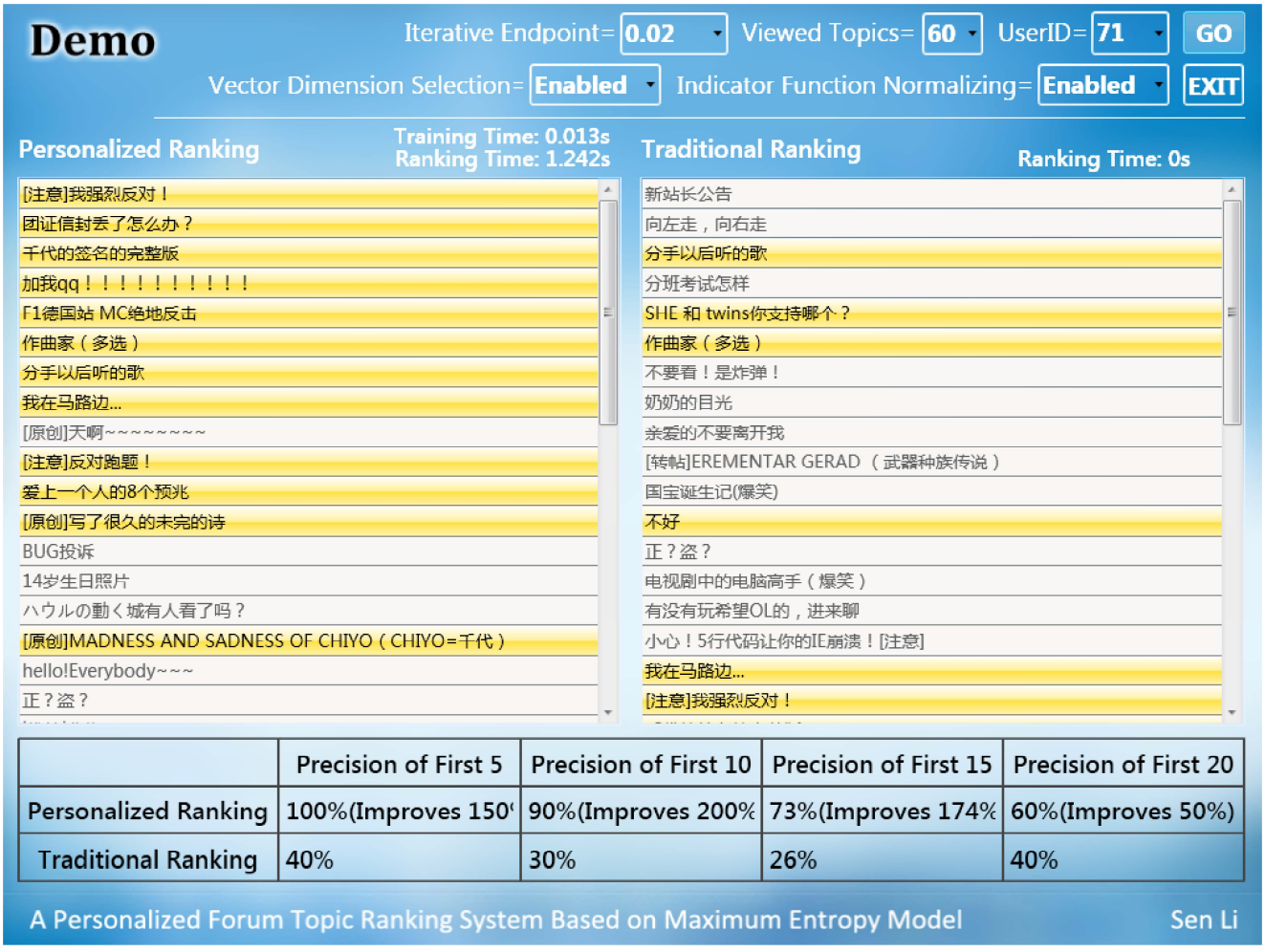
	Precision of First 5	Precision of First 10	Precision of First 15	Precision of First 20
Personalized Ranking	80%(Improves 100%	90%(Improves 200%	80%(Improves 199%	65%(Improves 62%)
Traditional Ranking	40%	30%	26%	40%



	Precision of First 5	Precision of First 10	Precision of First 15	Precision of First 20
Personalized Ranking	100%(Improves 150'	90%(Improves 200%	80%(Improves 199%	70%(Improves 75%)
Traditional Ranking	40%	30%	26%	40%



	Precision of First 5	Precision of First 10	Precision of First 15	Precision of First 20
Personalized Ranking	100%(Improves 150	90%(Improves 200%	73%(Improves 174%	65%(Improves 62%)
Traditional Ranking	40%	30%	26%	40%



	Precision of First 5	Precision of First 10	Precision of First 15	Precision of First 20
Personalized Ranking	100%(Improves 150	90%(Improves 200%	73%(Improves 174%	60%(Improves 50%)
Traditional Ranking	40%	30%	26%	40%